Incremental learning of Bayesian sensorimotor models: from low-level behaviours to large-scale structure of the environment

Julien Diard, Estelle Gilet, Eva Simonin, Pierre Bessiere

Connection Science

Incremental learning of Bayesian sensorimotor models: From low-level behaviours to large-scale structure of the environment

Julien Diard[(1),(*)], Estelle Gilet[(2)], Éva Simonin[(2)] and Pierre Bessière[(2)]

[(2)] *Laboratoire de Psychologie et NeuroCognition CNRS*
Université Pierre Mendès France, BSHM, BP 47
38040 Grenoble Cedex 9 – France
Tel: (+33) 4 76 82 58 93 – Fax: (+33) 4 76 82 78 34
`http://web.upmf-grenoble.fr/LPNC/`
Julien.Diard@upmf-grenoble.fr

[(2)] *Laboratoire d'Informatique de Grenoble - CNRS / INRIA Rhône-Alpes*
CNRS - Grenoble Universités, 655 avenue de l'Europe
38330 Montbonnot Saint Martin - France
Tel: (+33) 4 76 61 55 09 – Fax: (+33) 4 76 61 52 10
`http://www-laplace.imag.fr/`
estelle.gilet@gmail.com, evasimonin@yahoo.fr, Pierre.Bessiere@inrialpes.fr

(Received $$$; final version received $$$)
Roughly 9000 words.

Abstract

This paper concerns the incremental learning of hierarchies of representations of space in artificial or natural cognitive systems. We propose a mathematical formalism for defining space representations (Bayesian Maps) and modelling their interaction in hierarchies of representations (Sensorimotor Interaction operator).

We illustrate our formalism with a robotic experiment. Starting from a model based on the proximity to obstacles, we learn a new one related to the direction of the light source. It provides new behaviours, like phototaxis and photophobia. We then combine these two maps so as to identify parts of the environment where the way the two modalities interact is recognizable. This classification is a basis for learning a higher-level of abstraction map, that describes the large scale structure of the environment.

In the final model, the perception-action cycle is modelled by a hierarchy of sensorimotor models of increasing time and space scales, which provide navigation strategies of increasing complexities.

Keywords: space representation; navigation; Bayesian modelling; learning; behaviour

*Corresponding author: `Julien.Diard@upmf-grenoble.fr`

1 Introduction

1.1 *Context*

This paper concerns the domain of space representation and navigation, both from a mobile robotics perspective and from a cognitive science perspective.

In the engineering community, most approaches to robot mapping and autonomous navigation have focused on Bayesian, or probabilistic models. These provide a formal framework for treating uncertainties arising from the incompleteness of the models. The most successful methods are now able to map large structured indoor environments using probabilistic models. Unfortunately, they rely on large-scale, fine-grained, monolithic maps which are difficult to acquire and maintain. Such approaches, because of the difficulty of this task, are generally tailored toward one specific use of maps, such as localization, planning, or simultaneous localization and mapping. For instance, on the one hand, Markov localization [2] and Kalman Filters [16] have focused on the localization process, either using metric maps such as occupancy grids [26], or using topological maps [21], or even hybrid representations [25, 27]. On the other hand, many planning techniques have been developed in the context of probabilistic models of the environment [22, 10]. Behavior generation and task solving are seldom central to these approaches.

From a bio-mimetic robotics perspective, it appears obvious that a global, complex, large-scale model is not the starting point of the acquisition of representations of space [13]. Therefore, a few robotic approaches, integrating insights from biology, rather start from low-level behaviours and representations, and then try to combine them so as to obtain large-scale representations [3, 5, 13, 12, 30]. Indeed, the study of human and animal navigation capabilities assumes, right from the start of its analysis, that navigation is hierarchical in nature, as can be assessed experimentally [31]. Some of the bio-mimetic robotic space representation models [13, 12] implement hierarchical models of space representation from the behavioral biology literature [28, 6].

These hierarchies of models proposed have several aspects: they are hierarchies of increasing navigation skills, but also of increasing scale of the represented environments, of increasing time scale of the associated movements, and of increasing complexity of representations. This last aspect means that global topologic representations, which are simple, come at a lower level than global metric representations, which are arguably more complex to build and manipulate.

For instance, in recent works, Jacobs and Schenk propose the Parallel Map Theory (PMT) [8, 9], in which a study of neuroanatomical areas, which are phylogenetically equivalent across different species, helps formulate hypotheses about common hierarchies of representations of space. Redish and Touretzky [19] discuss possible neuroanatomical structures where different layers of space representations might be located, focusing on rat hippocampal and parahippocampal areas. In other words, these works propose models of how the different layers in the above theories might be implemented in the central nervous

system. Finally, Wang and Spelke [32, 33], taking insight from these theories of animal navigation, propose a three-component hierarchical model of human navigation.

1.2 *Problem specification*

However, to the best of our knowledge, most models of animal navigation from the behavioral biology and psychology literature are conceptual models, in the sense that they are not associated with complete operative mathematical definitions.

That is true for instance of the PMT mentioned above, or of the Wang and Spelke model of human navigation. Exceptions are found in the bio-inspired cognitive robotics literature [13], but, even there, the question of how different subsystems of a hierarchy of models can exchange information in a principled manner is still left as an open issue. In other words, most existing models of navigation inspired by biology describe hierarchies by identifying individual layers, but do not address the problem of how these layers are related. They usually assume that a supervisor subsystem is responsible for selecting the interaction between individual components, but rarely describe the way this supervisor could work, or even discuss its plausibility (*e.g.* the reference frame selection subsystem of Redish & Touretzky [19]).

Therefore, two challenges are to be tackled. The first one is to define models of how layers of space representations are articulated. This is the general context of our contribution. The second challenge is the study of how hierarchies of models can be learned experimentally by sensorimotor interaction with the environment. This is the focus of this paper. We firstly present a mathematical framework for building hierarchies of probabilistic models of space, based on Bayesian modelling and inference. Secondly, we illustrate this model in an experimental scenario, where a mobile robot incrementally learns sensorimotor models of its interaction with the environment.

1.3 *Proposed approach*

We use the Bayesian Map formalism (BM), which is a theoretical Bayesian framework for hierarchical space representation. We have previously argued [4] that the Bayesian Map formalism is a possible marriage between, on the one hand, "hierarchical bio-inspired models" and, on the other hand, "classical probabilistic mapping solutions". BMs are probability distributions over joint spaces of sensory and motor variables, which serve to compute particular probability distributions over motor variables, so as to provide behaviours.

BMs are also building blocks of hierarchies of representations: they can be put together using Bayesian operators, which take BMs as inputs and output new BMs. For instance, we previously defined the Superposition operator, which allows to merge BMs which describe the same physical space, but use different sensory modalities [5]. The Abstraction operator allows the building of hierarchies of maps: each location in the large-scale BM represents a lower-level,

underlying input BM [3].

In this work, we are concerned with the incremental learning of a hierarchy of BMs, built using a new operator for combining BMs, which we call the *Sensorimotor Interaction* operator. In this operator, applying a behaviour from one BM yields an identifiable effect on the internal variable of another BM. In other words, one BM based on one sensory modality acts as an observer, while another BM, based on another sensory modality, performs a navigation task.

Section 2 presents the mathematical definition of the BM concept and of the Sensorimotor Interaction operator. Section 3 briefly details the experimental platform we used: the Koala robot and the chosen environment.

We then turn to the description of the robotic experiments we carried out. Section 4 presents the two low-level BMs we defined, which are the first two blocks of our hierarchy. The first BM, BM_{prox}, allows the Koala robot to navigate (avoid walls, follow walls), based on proximity sensors. It is the result of a previous experiment [23], which we briefly summarize. Given this map, we then show how to learn a new map, BM_{light}, based on another sensory modality, using the light sensors of the Koala. Section 5 describes how we applied the Sensorimotor Interaction operator on BM_{prox} and BM_{light} so as to obtain BM_{arena}, the high-level BM.

2 Bayesian Maps and their sensorimotor interaction: mathematical definitions

In the BM formalism, maps are probabilistic models that provide navigation resources in the form of behaviours. These behaviours are formally defined by probabilistic distributions computed from the map. In this section, we present the formal definition of the BM concept and of the Sensorimotor Interaction of Bayesian Maps operator.

2.1 *Bayesian Maps*

A Bayesian Map c is a probabilistic model that defines a joint probability distribution $P(P\ L_t\ L_{t+\Delta t}\ A \mid c)$, where:

- P is a perception variable (the robot reads its values from physical sensors or lower-level variables),
- L_t is a location variable at time t,
- $L_{t+\Delta t}$ is a variable having the same domain than L_t, but at time $t + \Delta t$,
- and A is an action variable (the robot writes commands on this variable).

2.1.1 *Decomposition*

In order to be defined, the joint probability distribution $P(P\ L_t\ L_{t+\Delta t}\ A \mid c)$ is decomposed into a product of simpler terms. This choice of decomposition is not constrained: various probabilistic dependency structures are possible.

For example, the BM presented in section 4.1 follows the dependency structure defined in the Markov localization framework:

$$P(P\ L_t\ L_{t+\Delta t}\ A \mid c) = P(L_t \mid c)P(A \mid c)P(P \mid L_t\ c)P(L_{t+\Delta t} \mid A\ L_t\ c)\ .$$

In the following, $P(P \mid L_t\ c)$ is referred to as the *sensor model*, because it describes what should be read on the sensors given the current location. The term $P(L_{t+\Delta t} \mid A\ L_t\ c)$ is the *prediction model*, because it describes what location the robot should arrive at, given the past location and action.

Another possible decomposition is illustrated in section 4.2, where the probabilistic dependency structure is defined as:

$$P(P\ L_t\ L_{t+\Delta t}\ A \mid c) = P(P \mid c)P(A \mid c)P(L_t \mid P\ c)P(L_{t+\Delta t} \mid A\ L_t\ c)\ .$$

While very close to the Markov Localization probabilistic dependency structure, note that the sensor model $P(P \mid L_t\ c)$ is here replaced by $P(L_t \mid P\ c)$, the inverse relation, which corresponds to the probability distribution over the possible locations given the sensor readings. We call this term a *localization model*.

Finally, in our third BM example (see section 5.3), another decomposition is illustrated, where the action variable in used to refine the sensor model into $P(P \mid A\ L_t\ c)$, which we refer to as the *action-oriented sensor model*:

$$P(P\ L_t\ L_{t+\Delta t}\ A \mid c) = P(L_t \mid c)P(A \mid c)P(P \mid A\ L_t\ c)P(L_{t+\Delta t} \mid A\ L_t\ c)\ .$$

2.1.2 *Parametric forms*

We now need to define parametric forms for the terms in the chosen probabilistic dependency structure. The definition of parametric forms is not constrained. In the examples in this paper, we use Gaussian probability distributions, Dirac functions, Conditional Probability Tables (CPT) or uniform probability distributions.

The free parameters of these forms are experimentally identified by learning mechanisms. For instance, the *prediction model* $P(L_{t+\Delta t} \mid A\ L_t\ c)$ parameters can be obtained using the sensorimotor data recorded while the robot is applying a behaviour (initial exploratory behaviour or behaviours that arise from another BM). Indeed, at each time step, the location and the action of the robot are recorded. Thus, for each $\langle A, L_t \rangle$ pair, and assuming a Gaussian prediction model, the mean and variance of the robot location at the next time step $t + \Delta t$ can be computed using the data history.

2.1.3 *Behavior generation*

Once the joint probability distribution $P(P\ L_t\ L_{t+\Delta t}\ A \mid c)$ is specified and all the parameters are either learned of specified by hand, then the BM c is fully defined. It can then be used to solve navigation tasks, using Bayesian inference.

We define a navigation task by a probabilistic term to be computed, which we call a *question*: in the remainder of this paper, these tasks consist in trying to reach some goal in the internal space $L_{t+\Delta t}$. This goal is chosen by the programmer. To solve such a navigation task, the following question is asked to the BM c:

$$P(A \mid [P = p]\ [L_{t+\Delta t} = l_2]\ c)\ .$$

In other words: given its current sensor readings p and goal location l_2, what is the action the robot needs to apply? At each time step, the probability distribution is computed and an action is drawn at random according to it. In our examples, thanks to simple internal spaces $L_{t+\Delta t}$, the robot is able reach the goal l_2 using this simple action selection strategy.

We apply Bayesian inference to answer the question:

$$\begin{aligned} P(A \mid [P = p]\ [L_{t+\Delta t} = l_2]\ c) &= \frac{\sum_{L_t} P([P = p]\ A\ L_t\ [L_{t+\Delta t} = l_2]\ c)}{P([P = p]\ [L_{t+\Delta t} = l_2]\ c))} \\ &\propto \sum_{L_t} P([P = p]\ A\ L_t\ [L_{t+\Delta t} = l_2]\ c)\ . \end{aligned}$$

In our examples, thanks to the chosen decompositions, the summation over L_t can actually be approximated by a two-step inference algorithm. We first draw a value l_1 for L_t according either to the localization model $P(L_t \mid [P = p]\ c)$, or by "inverting" the sensor model $P([P = p] \mid L_t\ c)$ or the action-oriented sensor model $P([P = p] \mid A\ L_t\ c)$ [1]. Having drawn l_1, we use it for computing $P(A \mid [L_t = l_1]\ [L_{t+\Delta t} = l_2]\ c)$, by, here again, "inverting" the known prediction model $P([L_t = l_1] \mid A\ [L_{t+\Delta t} = l_2]\ c)$.

Having computed the probability distribution over A, we can now draw at random according to this distribution an action to perform to move closer to the goal. This action is executed by the robot to move in the environment: the robot is therefore applying a behaviour.

We therefore call *behaviours* questions of the form:

$$P(A \mid [P = p]\ [L_{t+\Delta t} = l_2]\ c) \text{ or } P(A \mid [L_t = l_1]\ [L_{t+\Delta t} = l_2]\ c)\ .$$

2.2 *Sensorimotor interaction of Bayesian Maps*

Having defined the Bayesian Map concept, we now turn to defining the Sensorimotor Interaction of Bayesian Maps operator.

Let c^1 and c^2 be two BMs, that deal respectively with variables P^1, L^1_t, $L^1_{t+\Delta t}$, A and P^2, L^2_t, $L^2_{t+\Delta t}$, A. They therefore are defined respectively by the joint probability distributions $P(P^1\ L^1_t\ L^1_{t+\Delta t}\ A \mid c^1)$ and $P(P^2\ L^2_t\ L^2_{t+\Delta t}\ A \mid c^2)$. Note that the action variable A is shared by the two models c^1 and c^2.

Let us also assume that c^1 and c^2 describe the same part of the environment of the robot, but do so in different terms (have different internal spaces L^1_t and L^2_t).

[1] In this last case, we use the last computed value of A, as it is precisely the goal of the current inference.

2.2.1 *Identification of large-scale zones: high-level action-oriented sensor model*

Each of these maps provide several behaviours so as to navigate relatively to the aspect of the part of the environment they describe. The behaviours are exclusive, in the sense that only one can be applied at a given time by the robot. Indeed, they both share the control variable A. While applying one behaviour, the corresponding BM is used to update its internal representation, given the corresponding sensory input, so as to decide what action should be made for reaching the goal. This leaves the other BM "free" to act as an observer. In other words, applying a behaviour from one of the BMs may have an identifiable effect on the internal variable of the other BM. For instance, the probability distribution over L_t^2, computed by c^2, could be stable over time, when some behaviour from c^1 is applied in some part of the environment. This is the knowledge we will exploit and experimentally learn.

Without loss of generality, let us assume that one behaviour from Beh^1, the set of behaviours from c^1, is applied, while its effect is observed by c^2. Let Env be the set of the parts of the environment where we will identify stable relations between the two maps. We will experimentally learn the following probability distribution:

$$P(L_t^2 \mid Beh^1 \; Env) \; . \tag{1}$$

Given this probability distribution, environment recognition can then be carried out by computing the term $P(Env \mid L_t^2 \; Beh^1)$ when a behaviour from c^1 is applied. Assuming uniform probability distribution priors for $P(L_t^2)$, and $P(Beh^1)$ yields that each of these term is computed by "inverting" the learned models of equation (1):

$$P(Env \mid L_t^2 \; Beh^1) \quad \propto \quad P(L_t^2 \mid Beh^1 \; Env) \; .$$

2.2.2 *Transitions between large-scale zones: high-level prediction term*

We now show how the Env variable is a basis for learning the large-scale structure of the arena.

Indeed, we can now learn the effect that the application of behaviours has on the temporal evolution of the Env variable. We thus define two instances of this variable at different time steps, Env_t and $Env_{t+\Delta t}$, and are interested in the prediction term $P(Env_{t+\Delta t} \mid Beh^1 \; Env_t)$: given the current recognized environment type, and given the behaviour which is applied, what is the probability distribution over environment types to be recognized in the future?

Learning this prediction term can easily be done by applying behaviours and building the experimental history of environment types recognized along the observed trajectory.

2.2.3 *Putting it all together: high-level Bayesian Map*

The trick to building the high-level BM, is to recognize that $P(L_t^2 \mid Env_t \; Beh^1)$ is a high-level action-oriented sensor model: Env_t is a coarse-grained location

variable, L_t^2, while it was a lower-level internal variable, is now a perception variable, and Beh^1, while it was defining behaviours, is now an action variable. $P(Env_{t+\Delta t} \mid Env_t\ Beh^1)$ is thus a high-level prediction model. We therefore can define a high-level BM that encapsulates these probability distributions.

More formally, we mathematically define the *Sensorimotor Interaction* operator as the operator which takes two BMs c^1 and c^2 as inputs, and outputs a new BM c^3, provided c^1 and c^2 share their action variable A [2]. c^3 is defined as follows:

- its perception variable is L_t^2;
- its internal variable at time t and $t + \Delta t$ are Env_t and $Env_{t+\Delta t}$, respectively;
- its action variable is Beh^1, the set of behaviours from c^1.

The joint probability distribution defining the Bayesian Map c^3 is:

$$\begin{aligned} & P(L_t^2\ Env_t\ Env_{t+\Delta t}\ Beh^1 \mid c^3) \\ = \quad & P(L_t^2 \mid c^3) P(Beh^1 \mid c^3) P(Env_t \mid L_t^2\ Beh^1\ c^3) \\ & P(Env_{t+\Delta t} \mid Beh^1\ Env_t\ c^3)\ . \end{aligned}$$

The terms $P(L_t^2 \mid c^3)$ and $P(Beh^1 \mid c^3)$ are defined by assuming uniform probability distributions. The action-oriented sensor model $P(Env_t \mid L_t^2\ Beh^1\ c^3)$ and the prediction model $P(Env_{t+\Delta t} \mid Beh^1\ Env_t\ c^3)$ are experimentally learned, as defined in the two previous sections.

Therefore, c^3 is fully defined, and can be used to define large-scale behaviours or policies, by computing, thanks to Bayesian inference:

$$P(Beh^1 \mid Env_t\ Env_{t+\Delta t}\ c^3)\ . \tag{2}$$

Given the current recognized environment type, and the environment type to be reached, what low-level behaviour should be chosen?

3 Experimental platform

`Figure 1 about here`

The experiment we describe here has been carried out on a Koala robot (Figure 1), which is a mid-size mobile robot (the base is roughly 30 × 30 cm).

[2] Actually, this operator outputs two symmetrical variants: one where L_t^2 is observed during the application of a behaviour from Beh^1, and one where L_t^1 is observed during the application of a behaviour from Beh^2. In this paper, we choose to focus on the former.

This robot is equipped with six wheels. There are two motors, each of which controls three wheels: the left motor controls the wheels on the left side, the right motor controls the wheels on the right side. As motors are controlled independently, the robot can turn on the spot, much like a tank. We command the robot by its rotation and translation speed (respectively *Rot* and $Trans$). The Koala is equipped with 16 infrared sensors. Each sensor can measure the ambient luminosity ($Lm_0, \ldots, Lm_{15}$) or the proximity to the nearest obstacle ($Px_0, \ldots, Px_{15}$). In fact, the latter is an information on the distance to the obstacle within 25 cm which depends a lot of the nature (orientation, colour, material, etc.) of the obstacle and the ambient luminosity.

The experiment took place in a large hall (roughly 10 × 5 m) without windows, on a carpeted floor. The environment of the robot consisted mainly of white boards or blue boxes, slightly taller than the robot, and which could be set in any configuration easily (see Figures 7 to 10 for examples). We also used a volleyball as a dynamic obstacle. The hall was in the dark, so that we could control that the main light source would be our spotlight.

4 Low-level Bayesian Maps: BM_{prox} and BM_{light}

In this section, we present robotic experiments where the Koala learns two BMs: BM_{prox} and BM_{light}. In each case, the scenario of the experiments is as follows.

The Koala is first given an initial behaviour and is asked to apply it in the environment. While it performs this behaviour, the sensory inputs and motor commands are recorded. These are the observations needed to learn the effect of an action on the perceptions of the robot. These observations are then used to build internal representations of the environment, in the form of BMs.

Actions can therefore be chosen so as to get closer to goals in the internal representation. In order to solve several navigation tasks, we will define several behaviours using the learned model.

4.1 *Proximetry-based Bayesian Map:* BM_{prox}

4.1.1 *Specification of the model*

In Section 2, we have given the general outline of our probabilistic model. Here it is instantiated to match the features of the Koala robot.

The perception variable P of BM_{prox} is the conjunction of 16 variables $Px_0, \ldots, Px_{15}$ corresponding to the 16 proximity sensors of the robot. Each variable has a value between 0 and 1023. The closer the obstacle is to the sensor, the higher the value.

The action variable A represents the different rotation speeds the robot can apply. Indeed, in this experiment we have chosen to set the translation variable to a fixed value, so that the only degree of freedom we control is the rotation speed. The action variable is therefore *Rot*, which can take three different values: -1 for turning to the left, +1 for turning to the right, and 0 for not turning.

Figure 2 about here

The location variables at time t L_t and $t+\Delta t$ $L_{t+\Delta t}$ have the same domain. In BM_{prox}, we assume that the robot sees only one obstacle at a time. A location is defined by the proximity and the angle of the robot to this obstacle. Thus, the location variable at time t is a pair of variables: Dir_t, for the direction of the obstacle and $Prox_t$, for its proximity. The Dir_t variable has 12 different integer values, from -5 to 6 (Figure 2). $Prox_t$ has 3 different values: $Prox_t = 0$ (respectively 1, 2) when the robot center is roughly 44 cm (respectively 35, 24) from the wall.

The variables being now instantiated, the joint probability distribution of BM_{prox} takes the form:

$$\begin{aligned} & P(Dir_t\ Prox_t\ Rot\ Px_0 \dots Px_{15}\ Dir_{t+\Delta t}\ Prox_{t+\Delta t} \mid BM_{prox}) \\ = \quad & P(Dir_t\ Prox_t \mid BM_{prox}) P(Rot \mid BM_{prox}) \\ & \prod_i P(Px_i \mid Dir_t\ Prox_t\ BM_{prox}) \\ & P(Dir_{t+\Delta t}\ Prox_{t+\Delta t} \mid Dir_t\ Prox_t\ Rot\ BM_{prox})\ . \end{aligned}$$

The 16 $P(Px_i \mid Dir_t\ Prox_t\ BM_{prox})$ terms, which define the sensor model, are gaussian distributions defined by 16 μ_1^i, σ_1^i mean and variance functions. In our experiment, these were learned in a supervised manner, by putting the robot at a given position with respect to an obstacle and recording 50 sensor data.

4.1.2 *Initial behaviour and learning of the prediction term*

Figure 3 about here

The behaviour known initially by the robot is a "random" behaviour. This behaviour consists in drawing a value according to a uniform distribution for *Rot* and maintaining the corresponding motor command for one second, before drawing a new value. During the second when the motor command is maintained, the robot observes how this motor command makes *Dir* and *Prox* vary. To learn the *prediction term*, this initial behaviour is applied.

During 5 minutes, every 100 ms, the values of Dir_t, $Prox_t$ and *Rot* were recorded. In this data history, it is possible to know, for a given start location $\langle dir_t, prox_t \rangle$ and applied action rot, the location of the robot at the next time step. Indeed, $\langle dir_t, prox_t, rot \rangle$ are a 3-tuple in the data, and the next location $\langle dir_{t+\Delta t}, prox_{t+\Delta t} \rangle$ is given in the first following 3-tuple (with $\Delta t = 100\ ms$) in the data history. Thus, a set of values for L_t, A and $L_{t+\Delta t}$ can be obtained.

The different values of $L_{t+\Delta t}$ found for a given set of L_t and A are used to compute the μ_2 and σ_2 functions defining the $P(L_{t+\Delta t} \mid L_t\ A)$ probability distribution. As $L_{t+\Delta t}$ is actually a pair $\langle Dir_{t+\Delta t}, Prox_{t+\Delta t}\rangle$, the prediction term is a set of 2D gaussian distributions, one for each possible value of $\langle dir_t, prox_t, rot\rangle$ (Figure 3).

4.1.3 *Behavior generation and experimental results*

Once the sensor and prediction models are learned, the joint distribution is fully defined. The internal representation of the sensorimotor space of the robot is acquired, and the BM can be used to generate new behaviours. For instance, a "ball pushing" behaviour is defined by computing, at each time step, the term:

$$P(Rot \mid [Dir_t = d]\ [Prox_t = p]\ [Dir_{t+\Delta t} = 0]\ [Prox_{t+\Delta t} = 2]\ BM_{prox})\ . \quad (3)$$

What is the probability distribution over rotation speeds, given the current position relative to the obstacle, and given that the goal is to have it in front ($Dir_{t+\Delta t} = 0$) and near the robot ($Prox_{t+\Delta t} = 2$)? Once the equation (3) is computed, it can be used to draw values for *Rot* to be applied by the robot. The result is an easily-recognizable ball-pushing behaviour.

In a similar fashion, a "left wall following" behaviour is obtained by defining the terms

$$\begin{aligned}
&P\left(Rot \,\middle|\, \begin{array}{l}[Dir_t = d]\ [Prox_t = 0]\\ [Dir_{t+\Delta t} = -2]\ [Prox_{t+\Delta t} = 1]\ BM_{prox}\end{array}\right), \text{ if } Prox_t = 0,\\
&P\left(Rot \,\middle|\, \begin{array}{l}[Dir_t = d]\ [Prox_t = 1]\\ [Dir_{t+\Delta t} = -3]\ [Prox_{t+\Delta t} = 1]\ BM_{prox}\end{array}\right), \text{ if } Prox_t = 1,\\
&P\left(Rot \,\middle|\, \begin{array}{l}[Dir_t = d]\ [Prox_t = 2]\\ [Dir_{t+\Delta t} = -4]\ [Prox_{t+\Delta t} = 1]\ BM_{prox}\end{array}\right), \text{ if } Prox_t = 2\ .
\end{aligned}$$

To obtain this behaviour, we set three different goals, depending on the value of $Prox_t$.

A wide variety of behaviours is obtained in this manner: right wall-following, ball avoidance, ball orbiting and obstacle avoidance. We refer the interested reader to the original paper for more details on this experiment [23]. In the remainder of this paper, we will only use *FWLeft* and *FWRight* the obtained left and right wall following behaviours, respectively.

4.2 *Light sensing-based Bayesian Map:* BM_{light}

We now show how BM_{prox} can be used as a basis for learning other sensorimotor models of the environment. The first step consists in learning a BM dealing with the light sensing modality of the robot, named BM_{light}, using a behaviour given by BM_{prox}.

4.2.1 *Specification of the model*

Figure 4 about here

We first define the Bayesian model of BM_{light}, using the Bayesian Robot Programming (BRP) notation [15] (Figure 4). We summarize now the most salient aspects of BM_{light}.

Figure 5 about here

The variables of BM_{light} are the sensory variables related to the light sensors $Lm_0 \dots Lm_{15}$, the motor variable Rot, controlling the rotation speed of the robot, and internal variables Lum_t and $Lum_{t+\Delta t}$ which respectively describe the current and future orientation of the strongest light source of the environment, relative to the robot. Lum_t and $Lum_{t+\Delta t}$ both have the same domain: they can take integer values from 0 to 7 (Figure 5).

The term $P(Lum_t \mid Lm_0 \dots Lm_{15}\ BM_{light})$ is the probability distribution over the position of the light source, given the light sensor readings. This term is a Dirac function centered on a value given by a function f of the sensor values $Lm_0 \dots Lm_{15}$. The f function is a simple max function over the sensor values: for instance, if the most excited sensor is numbered 0 or 8 (front sensors), then the probability that $Lum_t = 4$ is 1, and 0 for values other than 4 (Figure 5).

Finally, the term $P(Lum_{t+\Delta t} \mid Lum_t\ Rot\ BM_{light})$ is the prediction term: given the current light source orientation, and given the current motor command, how will the light source orientation evolve in the near future? This term is defined as a set of Gaussian probability distribution, whose means and standard-deviations can be learned experimentally.

4.2.2 *Initial behaviour and learning of the prediction term*

Figure 6 about here

In practice, we have the robot navigate in the environment, using the obstacle avoidance behaviour given by BM_{prox}. At each time step, we record the values of Lum_t and Rot. This allows to build an array of triplets $\langle lum_t, rot, lum_{t+\Delta t} \rangle$

for different values of Δt. In our experiment, we observed that $\Delta t = 1s$ gave the best results. Given the array of data, it is easy to compute the means and standard deviations of the probability distributions for the prediction term (Figure 6).

4.2.3 *Behavior generation and experimental results*

Once the prediction termed is acquired, BM_{light} is fully defined. It can thus be used to obtain behaviours, by computing probabilistic terms of the form:

$$P(Rot \mid Lum_t \ [Lum_{t+\Delta t} = l_2] \ BM_{light}) \ . \tag{4}$$

In these terms, the value l_2 can be interpreted as the goal value to reach in the domain of the internal variable Lum_t.

Computing equation (4) amounts to "inverting" the prediction term. Indeed, Bayesian inference yields (derivation omitted):

$$\begin{aligned} & P(Rot \mid Lum_t \ [Lum_{t+\Delta t} = l_2] \ BM_{light}) \\ & \quad \propto \quad P([Lum_{t+\Delta t} = l_2] \mid Rot \ Lum_t \ BM_{light}) \ . \end{aligned}$$

Once the distribution is computed, drawing at random according to it gives a rotation speed value which is sent to the motors, thus displacing the robot in the environment [3]. We have obtained two behaviours from BM_{light}: phototaxis and photophobia.

Figure 7 about here

Phototaxis is the behaviour of turning toward and going to the light source. It can be defined by the following probabilistic term:

$$P(Rot \mid Lum_t \ [Lum_{t+\Delta t} = 4] \ BM_{light}) \ .$$

What should be the motor command given the current orientation to the light source and given that the goal is to have it in front of the robot ($Lum_{t+\Delta t} = 4$)?

Photophobia can be defined in a similar manner, by setting the goal to be $Lum_{t+\Delta t} = 7$, which describes a light source position to the rear of the robot. We have recorded typical trajectories followed by the robot when applying the obtained behaviours (Figure 7).

[3] In practice, for security reasons, we only apply behaviours from BM_{light} when no obstacle is sensed near the robot ($Prox = 0$), otherwise, the obstacle avoidance behaviour is applied.

5 High-level Bayesian Map: BM_{arena}

BM_{prox} and BM_{light} model different aspects of the environment. BM_{prox} uses an internal space composed of $\langle Dir_t, Prox_t \rangle$ in order to describe the relation with the nearest obstacle. BM_{light} relies on a single internal variable, Lum_t, so as to describe the relative orientation of the strongest light source of the environment. Each of these map provides several behaviours so as to navigate relatively to the aspect of the environment they describe.

Applying a behaviour from one of the BMs may have an identifiable effect on the internal variable of the other BM. For instance, the probability distribution over Lum_t could be stable over time, when some behaviour from BM_{prox} is applied in some part of the environment. This is the knowledge we exploit and experimentally learn.

In the following, we will focus on *FWLeft*, *FWRight*, the two behaviours provided by BM_{prox}:$Beh_{prox} = \{FWLeft, FWRight\}$.

5.1 *Action oriented sensor model for environment recognition*

The sensorimotor interaction between the two BMs can be explored and learned, in order to create categories in the environment. These categories denote subsets of the environment where the relationship between sensorimotor models is identifiable by learning, and recognizable by inference.

5.1.1 *Experiment: wall orientation*

Figure 8 about here

The first experimental example concerns a simple case where three types of environment are considered. E_0, E_1 and E_2 are types of environment with a long wall and a single light source. They differ by the orientation of the wall relative to the light source (Figure 8, top row).

In a learning phase, the robot applies the *FWLeft* behaviour, given by BM_{prox}, while Lum_t values, computed by BM_{light}, were recorded. This gives three probability distributions, one for each type of environment (Figure 8, bottom row):

$$P(Lum_t \mid [Beh_{prox} = FWLeft]\ [Env_t = E_i]), i \in \{0, 1, 2\}\ .$$

Once these three probability distributions are learned, we can place the robot alongside a wall, ask it to follow it, and recognize the angle of the wall relative

to the light source, by computing:

$$\begin{aligned} & P(Env_t \mid Lum_t\ [Beh_{prox} = FWLeft]) \\ & \quad \propto \quad P(Lum_t \mid [Beh_{prox} = FWLeft]\ Env_t)\ . \end{aligned} \tag{5}$$

Here again, the computation yields an inversion of the learned model. Once equation (5) is computed, values for Env_t can be drawn at random according to $P(Env_t \mid Lum_t\ [Beh_{prox} = FWLeft])$: this provides a classification over the recognized types of environment.

The average success rate in recognition is roughly 70%. Misclassifications arise due to fluctuations in orientation of the robot relative to the wall: the *FWLeft* behaviour is not very stable in this respect. Moreover, environment types E_1 and E_2 are rather difficult to discriminate, as they both amount to having the light coming from the right of the robot, which only has two different sensors on the side.

`Figure 9 about here`

We have added a fourth type of environment in this first experiment: F (Figure 9). In F, the wall is placed between the robot and the light source, so that the robot lies partially in the shadow cast by the wall. We apply the learning method and report here the environment recognition results, with four learned types of environment: E_0, E_1, E_2 and F.

When the robot is placed in F, it correctly recognizes it 86% of the time. Interestingly, when placed in E_2, it correctly recognizes it only 45% of the time, and it thinks it is in F 25% of the time, which is its biggest misclassification rate when in E_2. In other words, it is difficult to discriminate between E_2 and F for our robot, when applying a *FWLeft* behaviour in these two types of environment. Further investigation showed that, in F, because of the cast shadow and the fact that the robot is very close to the wall when applying *FWLeft*, the light source is actually perceived only on the right sensors of the robot. This is also the case in E_2: these two geometrically distinct types of environment are really quite similar from a sensorimotor point of view.

5.1.2 *Experiment: wall height*

This experimental observation led us to a variant of our experiment, where we chose two types of environment to learn and discriminate: H_0 and H_1 (Figure 9). In these two types of environment, the wall position, the light source position, and the initial robot position are the same: the wall is placed between the robot and the light source. The only difference is the height of the wall: in H_0 the wall is small, in H_1 the wall is tall.

Since this changes the length of the cast shadow, this makes these two types of environment very different for the robot from a light sensing point of view.

And indeed, after a learning phase similar to the above experiment, the robot is quite able to recognize the wall height, when following the wall, with a 100% recognition rate. This result is the opposite of the previous finding: here, two geometrically similar types of environment (in 2D) are very different from a sensorimotor point of view. We stress the similarity of the 2D projection of these types of environment, as 2D maps are currently the goal of most classical mapping techniques: these would here fail to differentiate between the two obstacles H_0 and H_1.

5.1.3 *Experiment: wall colour*

The final experiment of environment recognition we report here concerns environment types I_0, I_1 and I_2 which, as before, are similar geometrically. In these types of environment, the robot is placed between the light source and the wall, which varies in colour, texture and material. The wall is made of wood planks that are painted white in I_0, it is made of blue cardboard boxes in I_1, and it is made of planks on which we affixed grey carpet strips in I_2.

After a learning phase, we asked the robot to recognize the type of environment it was in: when placed in I_0 or I_2, the robot recognizes them with a very high success rate (90% and 100%, respectively). Indeed, the surface reflectance are so different that it makes these types of environment very different from a light sensing point of view. In I_0, the white walls reflect so much light that the strongest light source perceived by the robot comes from its left. In I_2, the carpet texture reflects the light in a diffuse manner, so that the light source is detected as coming from the right. This, again, shows how different these two types of environment look from the point of view of the robot, even though they would be equivalent for a 2D mapper.

However, when placed in I_1, the robot recognizes as being either in I_0 or I_2, indifferently (50% each). In I_1, the light source is sometimes detected on the right of the robot, and sometimes on the left. The learning of a Gaussian distribution over this bimodal set of events yields a very flat probability distribution, with a mean centered on intermediate values (on the front of the robot). This is a consequence of the fact that, in I_1, the assumption of a single, main light source is wrong. In I_1, there are two light sources that are perceived with roughly the same intensity: the real spotlight, and its reflectance on the wall.

As a consequence, the model for I_1, relying on an inadequate assumption, is always less probable than those of I_0 or I_2 in the recognition inference. Therefore I_1 is never correctly recognized. This opens up interesting perspectives on the detection of inaccurate assumptions about the environment, which will require further experiments to explore.

5.2 *Learning the high-level prediction term*

Figure 10 about here

Having learned to recognize some types of environment, we now turn to the learning of the prediction term that links these types of environment. We have defined a small arena in which the robot could navigate, and placed a strong light source near it (Figure 10). The walls were placed, relative to the light source, at angles which corresponded to some of the previously learned types of environment: E_0, E_1, E_2 (which, as noted previously, is very similar to F), and J. In J, the robot has the light source on its back side.

In the learning phase, we had the robot navigate in the arena using either the left or right wall following behaviour. During this navigation, the environment recognition mechanism provided values for the Env_t variable. This allowed to learn conditional probability tables for the prediction term, using histograms.

Table 1 about here

The obtained probability tables for

$$P(Env_{t+\Delta t} \mid [Env_t = E_0]\ [Beh = FWLeft])\ ,$$

with $\Delta t = 900\ ms$, show that some of the structure of the arena is very well identified (Table 1). For instance, starting from J, the robot has a very high probability of staying in J ($p = 149/165$). Moreover, when it changes of type of environment, it is three times more likely to enter E_2 ($p = 12/165$) than to miss E_2 and arrive directly in E_1 ($p = 4/165$). In all cases, the robot has never experienced a direct transition from J to E_0. This corresponds to the physical reality of the used arena.

However, not all transition probabilities are this adequate, as some suffer from poor discrimination between types of environment: for instance, when in E_0, a direct transition to E_2, which was intuitively expected, was never experienced by the robot. And indeed, a more precise investigation of the phenomenon showed that, when passing from E_0 to E_2, in the physical arena, the robot had to enter the shadow cast by the wall of E_2. Therefore, temporarily, only the rear sensors of the robot were lit, which indeed corresponds to the J type of environment. The robot thus "correctly" learned a non-zero probability transition from E_0 to J ($p = 7/80$), which is even high compared to the other learned possible transition (from E_0 to E_1, with $p = 2/80$). This last transition

corresponds to entering the cast shadow while having already started to turn to the right, so that the front left sensors are lit, as in E_1.

Table 2 about here

When the behaviour applied is *FWRight*, different conditional probability tables are learned using the same mechanism (Table 2).

5.3 *Large-scale navigation using environment recognition:* BM_{arena}

We now encapsulate all learned models about the large-scale structure of the arena into BM_{arena} (Figure 11).

Figure 11 about here

We describe the behaviour selection for large-scale navigation in BM_{arena} using an example. Suppose we want the robot to reach E_0 in the arena. We can, at each time step, compute the probability distribution over behaviours given the goal $Env_{t+\Delta t} = E_0$, and given the current recognized type of environment Env_t. Assume that the robot currently is in E_1. We thus compute:

$$\begin{aligned} P(Beh \mid [Env_t = E_1]\ [Env_{t+\Delta t} = E_0]\ BM_{arena}) \\ \propto \quad P([Env_{t+\Delta t} = E_0] \mid [Env_t = E_1]\ Beh\ BM_{arena})\ . \end{aligned}$$

A lookup on the probability tables of the transition term (Table 1 and Table 2) yields:

$$\begin{aligned} P([Beh = FWLeft] \mid [Env_t = E_1]\ [Env_{t+\Delta t} = E_0]\ BM_{arena}) &\propto 10/336 \\ P([Beh = FWRight] \mid [Env_t = E_1]\ [Env_{t+\Delta t} = E_0]\ BM_{arena}) &\propto 0\ . \end{aligned}$$

Therefore, the behaviour to be chosen for going from E_1 to E_0 is, with a probability of 1, *FWLeft*. This correctly corresponds to the physical reality of the arena (Figure 10).

Not all of the behaviour selection computations would lead to such a clear choice. For instance, if the robot is in E_1 and tries to reach J, the probability of applying *FWLeft* is $(5/336)/(5/336 + 6/274) \simeq 0.40$, and the probability of applying *FWRight* is $(6/274)/(5/336+6/274) \simeq 0.60$. Furthermore, a behaviour selection, by drawing randomly according to this probability distribution at

each time step, yields unstable navigations where the robot quickly alternates between low-level behaviours. A possible solution to this problem is to lower the frequency of the behaviour selection, so that it better matches the dynamic of the changing of types of environment in the chosen arena. In other words, this corresponds to choosing a Δt which is more relevant (in this experiment, we just fixed it a priori).

6 Summary and Discussion

In this work, we have presented the Sensorimotor Interaction of Bayesian Maps operator, provided its mathematical definition, and applied it in a proof-of-concept robotic experiment on the Koala mobile robot. In this experiment, the robot first learned low-level BMs related to proximetry and light sensing. It then identified how these two sensorimotor modalities interacted, by learning a high-level BM. The internal variable of this last BM distinguishes parts of the environment where low-level models interact in a recognizable way. This allows to both localize in a coarse-grained space representation, and identify the large-scale structure of the arena the robot navigated in. We have shown surprising experimental results, where the robot was able to recognize properties of the environment (like colour or height of walls) not using dedicated sensors, but instead exploiting knowledge about its sensory and motor interaction with the environment.

We now discuss some of the implications of our work in the context of cognitive modelling of navigation and space representation.

6.1 *Perception-action cycle models*

In robotics or biology, navigation is usually modeled using a three-step algorithm, in order to go from sensors to actuators [17]: perceive (from sensors to some abstract internal space), plan (inference in the internal space to compute a path to be followed), and finally, act (application of motor commands to try to perform the planned path). This is the classical "perceive-plan-act" model.

Our perception-action cycle is more complex. In particular, variables in the resulting hierarchy have different natures in different sub-models. For instance, in BM_{light}, Lum_t acts as an internal variable, whereas it is a perception variable in BM_{arena}. This makes the computation flows in our model more complicated: instead of a series of unidirectional computations from sensors to actuators, Bayesian Maps operate in a parallel, hierarchical and, when the Δt are not equal in all the architecture, asynchronous manner.

In that sense, parts of the model "observe" other parts as they perform computations. In the Abstraction operator, this concept was even applied in a more general manner, as abstract BMs observed whole sub-models (and not just some variables), and computed their respective relevance to describing current sensorimotor data. This is a specific feature of Bayesian inference: indeed,

setting values for variables is actually a side-product of computations; their main goal is to obtain probabilities of models.

6.2 *Breaking down complexity*

There is a second noteworthy property of hierarchical models that can be illustrated by our experiments. Hierarchical models, once they are defined, break down complexity. Hard problems in large multidimensional spaces are replaced by several problems of lower computational costs. In our experiment, the internal variables define a 8-dimensional space:

$$Lum_t, Lum_{t+\Delta t}, Dir_t, Dir_{t+\Delta t}, Prox_t, Prox_{t+\Delta t}, Env_t, Env_{t+\Delta t} \ .$$

This internal space is decomposed, in the final model, into two 2-dimensional spaces and one 4-dimensional space. This makes handling each piece easier.

For instance, consider planning. Each BM, in our formalism, defines a prediction model of the form $P(L_{t+\Delta t} \mid A\ L_t)$, over a simple internal space. Such a prediction model can be directly reversed so as to compute policies $P(A \mid L_t\ L_{t+\Delta t})$, because the internal space is smooth and allows such a greedy hill-climbing action selection policy. In more complicated internal spaces, because of possible local minima, the prediction term would have to be iterated over time, so as to predict results of *series of actions*, which is a computationally very expensive process. In our robotic experiments, such explicit planning was not required.

The same argument also applies to learning. It has been argued, in a non hierarchical context, that learning behaviours makes perceptual learning easier [29]. We complement this argument here in a hierarchical context. We have replaced a learning problem in a large parameter space by a sequence of learning problems in low-dimensional parameter spaces. We have also shown how learning BM_{prox} initially was a basis for subsequent navigation and exploration, for acquiring experimental data. These were used to identify parameters of BM_{light} terms.

Finally, once the two low-level BMs were defined, experimentally learning their sensorimotor interaction produced BM_{arena}. The incremental nature of our learning process was used to our advantage. Being already set and defined, low-level models and behaviours produced recognizable effect in the environment, some which translated complex properties of the environment, but that we could nevertheless identify in the small internal space of BM_{arena}.

6.3 *Sensorimotor interaction vs. multi-modal fusion*

Let us now compare our Sensorimotor Interaction operator with a purely perceptual model, the sensor fusion model (in the terminology of [15]; also known as weak fusion [14], sensory weighting model [34], naïve Bayes fusion or classifier [18]). In this model of multi-modal perception, each modality i first computes some estimate ϕ_i about a property of interest in the environment. All k estimates are then combined by some fusion operator f, to produce

$S = f(S_0, \ldots, S_k)$. We claim that such a model would not have been able to recover properties like the height or colour of walls as we did in our experiment. Indeed, recall that our Koala is only equipped with proximity sensors and light sensors. None of these, taken individually, would be able to measure either the height or the colour of obstacles. In other words, each ϕ_i would not be sufficiently informative, given a single sensory snapshot. It is only by exploiting the interaction over time between sensory modalities and behaviours that recovering such properties is possible. We believe these interactions are very rich of meaning, even very early in sensory processes, in agreement with biology findings relative to early crossmodal interactions [11, 24].

6.4 *Sensorimotor models and representation*

Finally, we wish to offer some remarks with respect to the symbol grounding problem and sensorimotor models [7]. It can be noted that our robot is never given any symbolic concept of colour or height of walls, and nevertheless obtains operative sensorimotor models of them. These are denoted with symbols (I_0, I_1, I_2) that refer to probability distributions. These are used then to recognize whether the robot is near an obstacle that is similar to I_0, or to I_1, or to I_2. We believe that this mechanism of assigning single labels to complex sensorimotor models, which are relevant to some class of environment, might be the basis for the categorization of perceptual and motor experience into concepts.

It might also be noted that the representations that "make sense" for the robot, *i.e.*the labels that are easily distinguishable in the high-level BM, are grounded in the sensorimotor experience of the robot, and may be very different from concepts the programmer might have expected to be relevant. For instance, our robot identifies some structure of the large-scale arena, in terms of locations (Env_t values) and behaviours for going from location to location, but this structure is not always directly related to the geometry of the arena. Some locations are the same to the robot, even though they have very different geometries (*e.g.* E_2 and F); some locations are very easily distinguished by the robot, even though they have the same geometry (*e.g.* H_0 and H_1); finally, some locations are tied to an obvious geometric reference in the environment (E_0, E_1 and E_2 relate to the bearing of the light source with respect to the wall). This makes it sometimes difficult to the experimenter to visually assess the zones of the environment where some location should be recognized, but, on the other hand, it ensures that locations are relevant to the robot.

6.5 *Perspectives for future work*

In our experiment, a number of issues about hierarchical models of navigation and space representations were raised.

One of the most difficult challenge in this respect is studying the effect of time steps in each model of the hierarchy. Indeed, in this work, we have determined values for the Δt of each BM empirically. Setting this parameter automatically is an open issue. Indeed, when Δt is too small, actions mostly do not have any

effect in the internal space, as they do not have time to produce effects. In other words, whatever the action, the environment, observed at this frequency, does not change. At the other extreme, when Δt is too small, actions do have effects, but they become less and less predictable. Imagine watching a regular motion picture at 1000 Hz or 0.1 Hz: it mostly is the same picture, with very infrequent changes, or the missing pieces are so many that it becomes very difficult to predict future events.

Fortunately, in the Bayesian framework, these two extreme scenarios have different results. When Δt is too small, the prediction term is mostly filled with Dirac probability distributions centered on the starting location. When Δt is too large, the prediction term is mostly filled with uniform probability distributions. This was the leverage we used, empirically, to set Δt in our BMs: we varied this Δt parameter until the learned distributions were different enough from Dirac and uniform probability distributions. Learning automatically a suitable Δt value, according to some measure, probably summarizing the entropy of learned distributions, is an intriguing track for future work.

Another, even more difficult challenge lies in automatizing the incremental learning process. In our experiments, all the learning processes were started, ended, and sequenced by the human operator. Such a clean cut separation between learning processes is a highly unlikely model of animal or human learning. However, in the learning of complex sensorimotor skills, it is probably the case that the whole sensorimotor space is not explored and experienced in one block, in all its multidimensionality. Synergies (in the sense of [1]) probably set a hierarchical structure, which, as was outlined previously, already makes the learning process easier. In some instances, sequences of learning of different dimensions of complex sensorimotor spaces are also known to exist (*e.g.*pre-babblers vocalizations in speech learning [20]). How our model can adapt to less supervised epochs of parameter identification is an open question.

Finally, from a broader point of view, we believe our model to be a candidate formalization of some hierarchical models of navigation found in the behavioral biology literature. We have outlined previously how most of these lacked detailed mathematical definitions. The BM formalism, along with probabilistic operators we have defined to express hierarchies of representations of space, can be viewed as suggestions for models in experimental psychology. They are tailored for properties that appear relevant to studying human or animal navigation: hierarchies, behaviour generation, sensorimotor modelling of the interaction with the environment. Thus, we depart from classical robotic control architecture. Finally, while our formalism constitutes a promising first step, obtaining experimentally testable predictions from it is an important challenge.

Acknowledgments

This work has been supported by the BIBA European project (FP5-IST-2001-32115), and the BACS European project (FP6-IST-027140). We would like to thanks the anonymous reviewers for their helpful commentaries.

References

[1] A. Berthoz. *The Brain's Sense of Movement*. Harvard University Press, Cambridge, MA, 2000.

[2] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the* 13^{th} *Nat. Conf. on Artificial Intelligence and the* 8^{th} *Innovative Applications of Artificial Intelligence Conf.*, pages 896–901, Menlo Park, August, 4–8 1996. AAAI Press / MIT Press.

[3] J. Diard, P. Bessière, and E. Mazer. Hierarchies of probabilistic models of navigation: the bayesian map and the abstraction operator. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA04)*, pages 3837–3842, New Orleans, LA, USA, 2004.

[4] J. Diard, P. Bessière, and E. Mazer. A theoretical comparison of probabilistic and biomimetic models of mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA04)*, pages 933–938, New Orleans, LA, USA, 2004.

[5] J. Diard, P. Bessière, and E. Mazer. Merging probabilistic models of navigation: the bayesian map and the superposition operator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05)*, pages 668–673, 2005.

[6] M. Franz and H. Mallot. Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30:133–153, 2000.

[7] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[8] L. F. Jacobs. The evolution of the cognitive map. *Brain, behavior and evolution*, 62:128–139, 2003.

[9] L. F. Jacobs and F. Schenk. Unpacking the cognitive map: the parallel map theory of hippocampal function. *Psychological Review*, 110(2):285–315, 2003.

[10] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[11] M. Kubovy and D. V. Valkenburg. Auditory and visual objects. *Cognition*, 80:97–126, 2001.

[12] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA04)*, pages 4845–4851, New Orleans, LA, USA, 2004.

[13] B. J. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1–2):191–233, 2000.

[14] M. S. Landy, L. T. Maloney, E. B. Johnston, and M. Young. Measurement and modeling of depth cue combination: in defense of weak fusion. *Vision Research*, 35(3):389–412, 1995.

[15] O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian robot programming. *Autonomous Robots*, 16(1):49–79, 2004.

[16] J. Leonard, H. Durrant-Whyte, and I. Cox. Dynamic map-building for an autonomous mobile robot. *The International Journal of Robotics Research*, 11(4):286–298, 1992.

[17] T. S. Levitt and D. T. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3):305–360, 1990.

[18] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[19] A. D. Redish and D. S. Touretzky. Cognitive maps beyond the hippocampus. *Hippocampus*, 7(1):15–35, 1997.

[20] J. Serkhane, J.-L. Schwartz, and P. Bessière. Building a talking baby robot : A contribution to the study of speech acquisition and evolution. *Interaction Studies*, 6(2):253–286, 2005.

[21] H. Shatkay and L. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI-97)*, pages 920–929, San Francisco, August, 23–29 1997. Morgan Kaufmann Publishers.

[22] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1080–1087, 1995.

[23] É. Simonin, J. Diard, and P. Bessière. Learning Bayesian models of sensorimotor interaction: from random exploration toward the discovery of new behaviors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05)*, pages 1226–1231, 2005.

[24] C. Spence and S. Squire. Multisensory integration: Maintaining the perception of synchrony. *Current Biology*, 13:R519–R521, 2003.

[25] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[26] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[27] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44:3–14, 2003.

[28] O. Trullier, S. Wiener, A. Berthoz, and J.-A. Meyer. Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544, 1997.

[29] P. F. M. J. Verschure, T. Voegtlin, and R. J. Douglas. Environmentally mediated synergy between perception and behaviour in mobile robots. *Nature*, 425:620–624, october 2003.

[30] A. C. Victorino and P. Rives. An hybrid representation well-adapted to the exploration of large scale indoors environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA04)*, pages 2930–2935, New Orleans, LA, USA, 2004.

[31] H. Voicu. Hierarchical cognitive maps. *Neural Networks*, 16:569–576, 2003.

[32] R. F. Wang and E. S. Spelke. Updating egocentric representations in human navigation. *Cognition*, pages 215–250, 2000.

[33] R. F. Wang and E. S. Spelke. Human spatial representation: insights from animals. *TRENDS in Cognitive Science*, 6(9):376–382, 2002.

[34] L. H. Zupan, D. M. Merfeld, and C. Darlot. Using sensory weighting to model the influence of canal, otolith and visual cues on spatial orientation and eye movements. *Biological Cybernetics*, 86:209–230, 2002.

	E_0	E_1	E_2	J
$P(Env_{t+\Delta t} \mid [Env_t = E_0][Beh = FWLeft])$	71/80	2/80	0	7/80
$P(Env_{t+\Delta t} \mid [Env_t = E_1][Beh = FWLeft])$	10/336	309/336	12/336	5/336
$P(Env_{t+\Delta t} \mid [Env_t = E_2][Beh = FWLeft])$	0	21/218	193/218	4/218
$P(Env_{t+\Delta t} \mid [Env_t = J][Beh = FWLeft])$	0	4/165	12/165	149/165

Table 1: Learned transition matrices for the term $P(Env_{t+\Delta t} \mid Env_t\ [Beh = FWLeft])$, for all values of Env_t. Probability distributions over the $Env_{t+\Delta t}$ variable are read in rows.

	E_0	E_1	E_2	J
$P(Env_{t+\Delta t} \mid [Env_t = E_0][Beh = FWRight])$	93/105	8/105	0	4/105
$P(Env_{t+\Delta t} \mid [Env_t = E_1][Beh = FWRight])$	0	263/274	5/274	6/274
$P(Env_{t+\Delta t} \mid [Env_t = E_2][Beh = FWRight])$	0	1/212	196/212	15/212
$P(Env_{t+\Delta t} \mid [Env_t = J][Beh = FWRight])$	12/208	3/208	10/208	183/208

Table 2: Learned transition matrices for the term $P(Env_{t+\Delta t} \mid Env_t\ [Beh = FWRight])$, for all values of Env_t. Probability distributions over the $Env_{t+\Delta t}$ variable are read in rows.

Figure 1: The Koala® mobile robot (K-Team company), next to a volleyball, for size reference. The red and green dots on top of the robot were not used in the experiments described here.

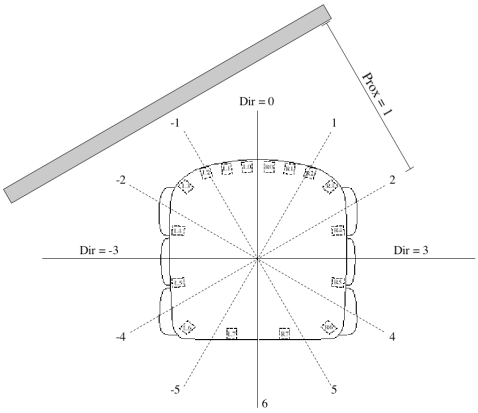

Figure 2: The *Dir* and *Prox* variables, on an overhead schema of the Koala robot.

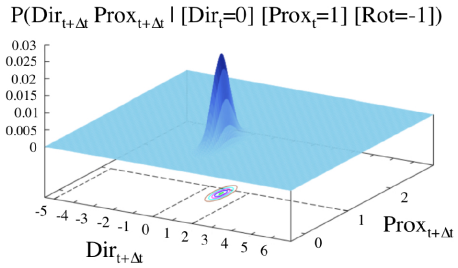

Figure 3: Example of probability distribution learned for the prediction term $P(Dir_{t+\Delta t}\ Prox_{t+\Delta t} \mid Dir_t\ Prox_t\ Rot\ BM_{prox})$, with $\Delta t = 900$ ms. The prediction term for the starting location $Dir_t = 0$, $Prox_t = 1$ and motor command $Rot = -1$. The plot shows that if the robot had the wall in front of it, at a medium distance, and it turned to the left ($Rot = -1$), then the wall would probably be on the right front side of the robot 900 ms later.

Program

Description

Specification

Relevant variables

$$
\begin{array}{lcl}
P & = & Lm_0 \wedge \ldots \wedge Lm_{15} \; ; \quad \mathcal{D}_P = \{0, \ldots, 1023\}^{16} \\
L_t & = & Lum_t \; ; \quad \mathcal{D}_{L_t} = \{0, \ldots, 7\} \\
L_{t+\Delta t} & = & Lum_{t+\Delta t} \; ; \quad \mathcal{D}_{L_{t+\Delta t}} = \{0, \ldots, 7\} \\
A & = & Rot \; ; \quad \mathcal{D}_A = \{-1, 0, 1\}
\end{array}
$$

Decomposition

$$
\begin{array}{l}
P(Lum_t \; Rot \; Lm_0 \ldots Lm_{15} \; Lum_{t+\Delta t}) \\
\quad = \quad P(Lm_0 \ldots Lm_{15}) P(Rot) P(Lum_t \mid Lm_0 \ldots Lm_{15}) \\
\qquad P(Lum_{t+\Delta t} \mid Lum_t \; Rot)
\end{array}
$$

Parametric forms

$$
\begin{array}{lcl}
P(Lm_i) & = & \mathbf{U}(Lm_i) \\
P(Rot) & = & \mathbf{U}(Rot) \\
P(Lum_t \mid Lm_0 \ldots Lm_{15}) & = & \boldsymbol{\delta}_{f(Lm_0 \ldots Lm_{15})}(Lum_t) \\
P(Lum_{t+\Delta t} \mid Lum_t \; Rot) & = & \mathbf{G}_{\substack{\mu(Lum_t, Rot), \\ \sigma(Lum_t, Rot)}}(Lum_{t+\Delta t})
\end{array}
$$

Identification : Learning of $P(Lum_{t+\Delta t} \mid Lum_t \; Rot)$

Question : definition of *behaviours*

$$
\begin{array}{l}
P(Rot \mid [Lum_t = l_1] \; [Lum_{t+\Delta t} = l_2]) \\
\quad = \quad \frac{1}{Z} \; P([Lum_{t+\Delta t} = l_2] \mid Rot \; [Lum_t = l_1])
\end{array}
$$

Figure 4: Bayesian Robotic Program summarizing BM_{light}. For clarity, the symbol BM_{light}, which should appear in every right-hand side of the above probabilistic terms, has been omitted. $\mathbf{U}(X)$ denotes uniform probability distributions over the variable X, $\boldsymbol{\delta}_f(X)$ denotes Dirac probability distributions, and $\mathbf{G}_{\mu,\sigma}(X)$ denotes Gaussian probability distributions of mean μ and variance σ.

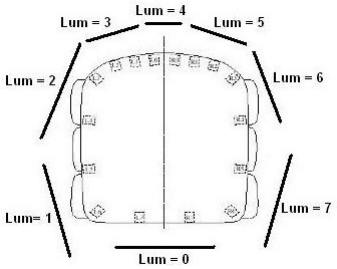

Figure 5: Overhead schema of the Koala robot, with the position of the sensors, and values for the Lum_t variable.

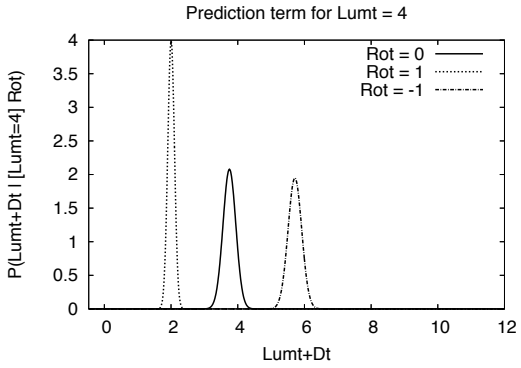

Figure 6: Some of the Gaussian probability distributions for the prediction term. These are $P(Lum_{t+\Delta t} \mid [Lum_t = 4]\ [Rot = x]\ BM_{light})$, with $x = 1$ (left curve), $x = 0$ (middle curve), and $x = -1$ (right curve). For example, the right curve encodes the knowledge that, if the light was in front ($Lum_t = 4$), and the robot turned to the left ($Rot = -1$), then the light will most likely be on the right of the robot in the near future (Gaussian distribution centered on $Lum_{t+\Delta t} = 6$, roughly).

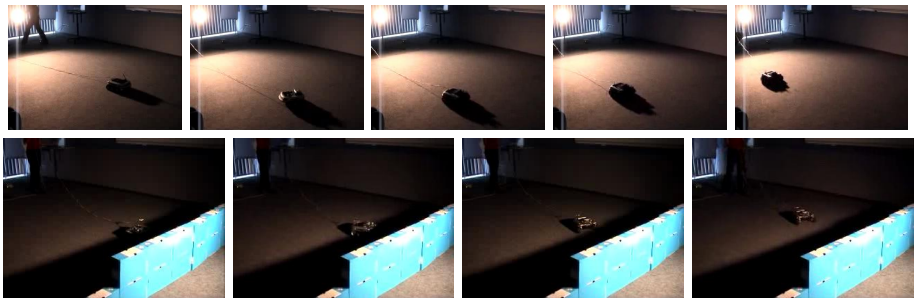

Figure 7: Pictures of the robot applying phototaxis (top row) and photophobia behaviour (bottom row).

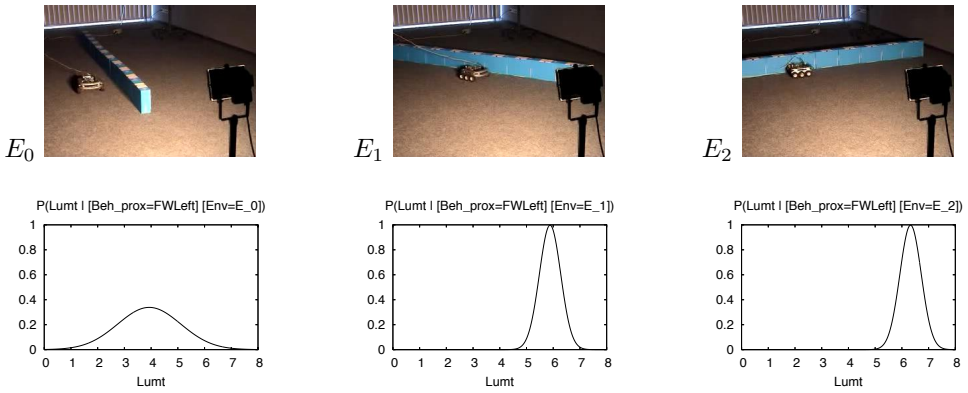

Figure 8: Top row: pictures of types of environment E_0, E_1, E_2; bottom row: learned probability distributions $P(Lum_t \mid [Beh_{prox} = FWLeft]\ [Env_t = E_i]), i \in \{0, 1, 2\}$. For example, the left most graph shows that, when following the wall, the robot mostly had the light source in front of it, which results in a Gaussian distribution centered on the value 4. The standard deviation of this distribution reflects the fluctuations in the orientation of the robot that result from rotations used to correct the alignment of the robot with the wall.

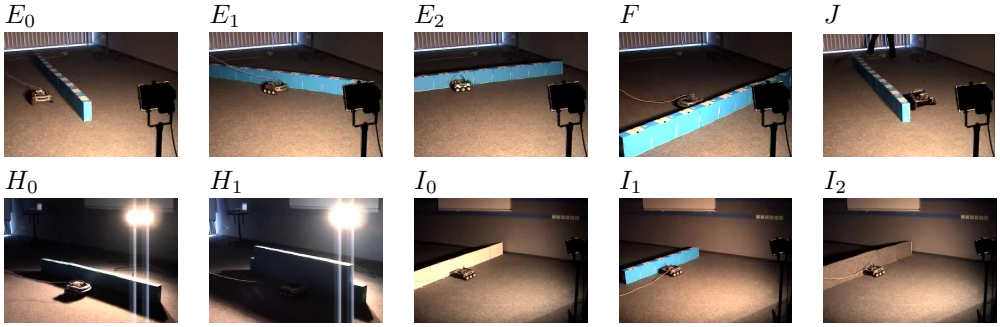

Figure 9: Environments E_0, E_1, E_2, F and J (top row). Environments H_0, H_1, I_0, I_1 and I_2 (bottom row).

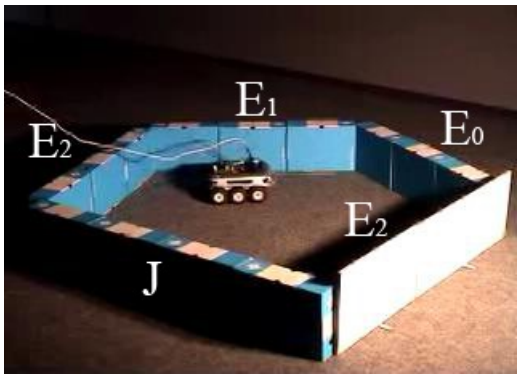

Figure 10: Arena made of the types of environment E_2, E_1, E_0 and J.

Program

Description

Specification

Relevant variables

$$
\begin{array}{lcl}
P & = & Lum_t \ ; \quad \mathcal{D}_{L_t} = \{0, \ldots, 7\} \\
L_t & = & Env_t \ ; \quad \mathcal{D}_{L_t} = \{E_0, E_1, E_2, J\} \\
L_{t+\Delta t} & = & Env_{t+\Delta t} \ ; \quad \mathcal{D}_{L_{t+\Delta t}} = \{E_0, E_1, E_2, J\} \\
A & = & Beh \ ; \quad \mathcal{D}_A = \{FWLeft, FWRight\}
\end{array}
$$

Decomposition

$$
\begin{array}{l}
P(Lum_t \ Dir_t \ Prox_t \ Env_t \ Env_{t+\Delta t} \ Beh) \\
\quad = \quad P(Lum_t \mid Beh \ Env_t) P(Env_t) P(Beh) P(Env_{t+\Delta t} \mid Beh \ Env_t)
\end{array}
$$

Parametric forms

$$
\begin{array}{lcl}
P(Env_t) & = & \mathbf{U}(Env_t) \\
P(Beh) & = & \mathbf{U}(Beh) \\
P(Lum_t \mid Beh \ Env_t) & = & \mathbf{G}_{\substack{\mu(Beh, \, Env_t), \\ \sigma(Beh, \, Env_t)}} (Lum_t) \\
P(Env_{t+\Delta t} \mid Beh \ Env_t) & = & \mathbf{CPT}
\end{array}
$$

Identification : Learning of $P(Lum_t \mid Beh \ Env_t), P(Env_{t+\Delta t} \mid Beh \ Env_t)$

Question : Large-scale navigation by selection of low-level behaviours

$$
P(Beh \mid Env_t \ Env_{t+\Delta t}) \quad \propto \quad P(Env_{t+\Delta t} \mid Env_t \ Beh)
$$

Figure 11: Summary of BM_{arena}. For clarity, the symbol BM_{arena}, which should appear in every right-hand side of the above probabilistic terms, has been omitted. CPT stands for "Conditional Probability Tables".